


REVISIONS			
REV	DESCRIPTION	DATE	APPROVED
1.0	Based on MC9090 Ext Power Control User Document	12/15/2011	Lasantha Jayathilake
<div>  MOTOROLA SOLUTIONS </div> <div>Enterprise Solutions</div>			
<div> <h1>MC9190 EXT POWER CONTROL</h1> <h2>USER DOCUMENT</h2> </div>			
		Page 1 OF 10	

TABLE OF CONTENTS

1.	INTRODUCTION	3
1.1	Purpose	3
2.	MC9190 EXTERNAL POWER PIN	4
2.1	Background	4
2.2	Theory of Operation	4
3.	DESIGN.....	7
3.1	Software Design	7
3.1.1	<i>Sample code to enable external power.....</i>	<i>7</i>
3.1.2	<i>State at EXP Open and Close.....</i>	<i>8</i>
3.1.3	<i>Boot up state.....</i>	<i>8</i>
3.1.4	<i>Suspend and Resume</i>	<i>9</i>
3.2	Test Application	9
3.2.1	<i>Open and Close buttons in the COM Block</i>	<i>9</i>
3.2.2	<i>Open and close buttons in the Power Block.....</i>	<i>9</i>
3.2.3	<i>Enable and Disable buttons in the Power block</i>	<i>9</i>
3.2.4	<i>Blank text box and the “X” button</i>	<i>9</i>
4.	NOTES	10

TABLE OF FIGURES

Figure 1 – MC9190 bottom port	4
Figure 2 - Block diagram.....	6
Figure 3 - Test Application	9

LIST OF TABLES

Table 1: MC9190 Bottom Port Pin-Outs	5
--	---

Introduction

1.1 Purpose

The purpose of this document is to describe the software design of MC919x External power pin control. This is a part of a project designing and building a custom ADP9000 accessory that can provide power to charge 3rd party USB accessories using MC919x battery.

2. MC9190 External Power Pin

2.1 Background

Motorola MC9190 hand held computer supports a series of accessories through its bottom electrical connector. This connector has number of communication ports and a power line to provide power to these external devices. Historically this power pin has been controlled by the communication port driver used to communicate to the accessory.

MC919x External Power Control Driver provides an extra interface that can be used to control the external power pin without affecting the communication port. This can be used with Custom hardware accessory ADP9000-120R to charge 3rd party USB devices from MC919x battery.

2.2 Theory of Operation

Mobile Computer Pin-Outs

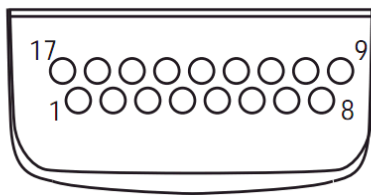


Figure 1 – MC9190 bottom port

PIN	Signal Name	Function
1	USB_GND	USB
2	USB_D_PLUS	USB
3	TXD	RS232C
4	RXD	RS232C
5	DCD	RS232C
6	RTS	RS232C
7	DSR	RS232C
8	GND	Ground, 2.5A max
9	RI	RS232C
10	CRADLE_DET	Grounded by cradle when in cradle
11	DTR	RS232C
12	Not	connected Not connected
13	POWER_IN	12V, 2.5A max
14	CTS	RS232C
15	USB_5V_DET	USB
16	USB_D_MINUS	USB
17	EXT_PWR_OUT	500mA max

Table 1: MC9190 Bottom Port Pin-Outs

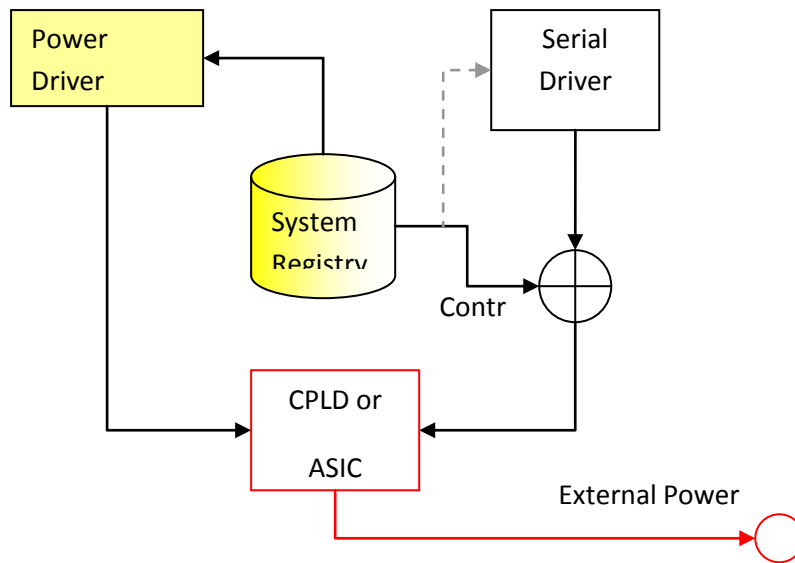


Figure 2 - Block diagram

External power pin is controlled by an ASIC and it can be controlled by writing to a hardware register. There is more than one software component that has access to this hardware. For this custom product we are looking at COM1 (serial printer port) only.

The solution is provided as a CAB installer and the user has to install it and do a manual warm boot. A clean boot will erase the external power driver and restore the device to previous state.

3. Design

3.1 Software Design

The new External power driver (EXP1) directly access the H/W and controls external power pin. The existing Serial Driver also accesses this pin but its activity can be controlled via system registry.

Here we have blocked COM1 controlling external power pin using the below registry entry.

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\Serial\DevPower]
    "PwrControlType"=dword:00000000
```

Then we load External power driver as a built in driver using below registry entry so that user application can open it using EXP1.

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\ExtPwrPin]
    "FriendlyName"="External Power Pin Driver"
    "Prefix"="EXP"
    "Dll"="ExtPwrPin.dll"
    "Index"=dword:00000001
    "Order"=dword:00000005
    "IClass"=MultiSZ: {A32942B7-920C-486b-B0E6-92A702A99B35}
```

We have below Device IO control exposed out for the application developer to use.

```
IOCTL_ENABLE_EXTPWRPIN
```

3.1.1 Sample code to enable external power

```
#include <winIoctl.h>

#define FILE_DEVICE_EXTERNALPOWERPIN_TOGGLER 0x00007000
#define IOCTL_ENABLE_EXTPWRPIN \
    CTL_CODE(FILE_DEVICE_EXTERNALPOWERPIN_TOGGLER, 0x801, \
    METHOD_BUFFERED, FILE_WRITE_ACCESS)

HANDLE g_hPwrPort=INVALID_HANDLE_VALUE;
BOOL bVal=TRUE; //Set True for Power ON

//open External power driver

g_hPwrPort = CreateFile (L"EXP1:", GENERIC_READ, FILE_SHARE_READ, \
    0, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
//If we have a valid handle, call DeviceIoControl
if(INVALID_HANDLE_VALUE != g_hPwrPort)
    DeviceIoControl(g_hPwrPort, IOCTL_ENABLE_EXTPWRPIN, \
    (LPVOID) &bVal, sizeof(BOOL), NULL, NULL, NULL, NULL);
```

Above code sample can be used to open the EXP1 and to set power pin to ON.

3.1.2 State at EXP Open and Close

To provide user more control to the power pin, below registry entries can be used to define its status at EXP1 Open and Close. This is useful in situations where the user expect to automatically reset the power when the he exit from the application.

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\ExtPwrPin]
    "OpenPowerState"=dword: 00000002
    "ClosePowerState"=dword: 00000000
```

dword = 0 - OFF (Default for ClosePowerState)
dword = 1 - ON
dword > 2 - UNCHANGE (Default for OpenPowerState)

The default state is defined as: when the user opens EXP1, it will not change the power state but when it is closed the power will go off. User can manipulate these entries such that the power goes ON at EXP1 open and goes OFF and EXP1 close so that the device IO control can be bypassed.

IMPORTANT

These settings will be loaded only at a system startup so the user need to warm boot the device after changing any of the above registry entries.

If ClosePowerState is set to 1, the device will set the External Power pin to ON after a warm boot. This happens due to device manager calling the device IO control at system startup which involves an EXP1 Open followed by an EXP1 Close. Similarly if OpenPowerState is set to 1, and ClosePowerState is set to "Unchange", the external power pin will stay on after a warm boot.

3.1.3 Boot up state

The state of the external power pin after Cold/Warm boot is controlled by the below registry entry.

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\ExtPwrPin]
    "BootPowerState"=dword: 00000000
BootPowerState = 0 - External pin is OFF at warm boot (Default)
BootPowerState > 1 - External pin is ON at warm boot
```

Any value greater than 1 will be considered as 1, and if this entry is not there the default (0) will be used.

IMPORTANT

These settings will be loaded only at a system startup so the user need to warm boot the device after changing any of the above registry entries.

To use this feature, EXP1 OpenPowerState and ClosePowerState entries has to be set to UNCHANGE (2). Device manager will call EXP open and Close at driver Init so it will be first set to OpenPowerState and then to ClosePowerState regardless of what is defined as BootPowerState.

3.1.4 Suspend and Resume

At suspend, the external power pin will go off and at resume will be restored to the original state.

3.2 Test Application

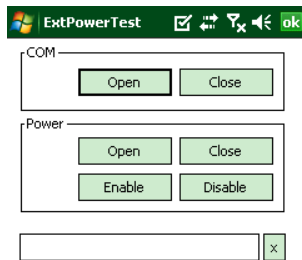


Figure 3 - Test Application

Above application provides a basic test platform for the EXP1 power driver. It is copied to MC9190 devices \Program Files\ExtPowerTest folder when the driver CAB file is installed.

3.2.1 Open and Close buttons in the COM Block

These two buttons will open and close the built in COM port COM1. When the power driver is installed it should not change the external power pin state.

3.2.2 Open and close buttons in the Power Block

There two buttons will open and close EXP1. With default settings this will not change the external power pin state but “Close” button will switch external power off if it is set to ON by “Enable” button.

3.2.3 Enable and Disable buttons in the Power block

These two buttons will call the device IO control to ON and OFF power to the external power pin.

3.2.4 Blank text box and the “X” button

The text box will display any error state, and the “X” button can be used to clear the text box.

4. Notes